



Topology optimization of unsteady flow problems using the lattice Boltzmann method

Nørgaard, Sebastian Arlund; Sigmund, Ole; Lazarov, Boyan Stefanov

Published in:
Journal of Computational Physics

Link to article, DOI:
[10.1016/j.jcp.2015.12.023](https://doi.org/10.1016/j.jcp.2015.12.023)

Publication date:
2016

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Nørgaard, S. A., Sigmund, O., & Lazarov, B. S. (2016). Topology optimization of unsteady flow problems using the lattice Boltzmann method. *Journal of Computational Physics*, 307, 291-307.
<https://doi.org/10.1016/j.jcp.2015.12.023>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Topology optimization of unsteady flow problems using the lattice Boltzmann method

Sebastian Nørgaard^{a,*}, Ole Sigmund^a, Boyan Lazarov^a

^a*Department of Mechanical Engineering, Technical University of Denmark, DK-2800 Lyngby, Denmark*

Abstract

This article demonstrates and discusses topology optimization for unsteady incompressible fluid flows. The fluid flows are simulated using the lattice Boltzmann method, and a partial bounceback model is implemented to model the transition between fluid and solid phases in the optimization problems. The optimization problem is solved with a gradient based method, and the design sensitivities are computed by solving the discrete adjoint problem. For moderate Reynolds number flows, it is demonstrated that topology optimization can successfully account for unsteady effects such as vortex shedding and time-varying boundary conditions. Such effects are relevant in several engineering applications, i.e. fluid pumps and control valves.

Keywords: Topology optimization, Unsteady flow, Lattice Boltzmann

1. Introduction

The lattice Boltzmann method (LBM) has gained increasing attention in recent years as a viable alternative to Navier-Stokes (NS) based methods for computational fluid dynamics. In this paper, we present a method for topology optimization of fluid domains subject to unsteady flow conditions using the LBM. The topology of the domain is represented using the density approach, in which each point of the discretized domain is assigned a value between 0 and 1, representing a fully solid and fluid node, respectively [1]. In contrast to previously published work on fluid domain optimization with the LBM, which has focused on steady-state formulations of the optimization problem, this work considers an unsteady flow formulation, and presents optimization problems in which unsteady effects have a significant influence on the optimized topologies.

The topology optimization methodology was originally developed as a design tool for structural mechanics [1, 2], and since its inception it has been applied to a variety of physical domains, including the optimal control of fluid flow problems. Borrvall and Petersson first applied topology optimization to Stokes flow problems in 2003 [3], and since then, additional studies have extended their work to laminar Navier-Stokes flow at moderate Reynolds number [4, 5], and to large scale problems solved using parallel computation [6]. Furthermore, there are numerous published studies applying the methodology to more complicated fluidic devices, such as fluid switches [4], and microfluidic mixers [7, 8].

In the majority of previous works, the flow solutions are approximated by numerically solving the NS equations. Recently, however, the LBM has become a popular alternative to traditional solvers. For a thorough introduction, the reader is referred to the books by Sukop and Thorne [9], and Wolf-Gladrow [10]. In contrast to the continuum assumption of the NS based approach, the starting point of the LBM is the Boltzmann transport equation of kinetic theory. Discretization of this equation yields an explicit time marching scheme which can compute approximate solutions to the incompressible NS equations for low Mach number flows. The method is attractive because it is algorithmically simple, lends itself well to

*Corresponding author

Email address: sebnorg@mek.dtu.dk (Sebastian Nørgaard)

parallel implementation, and is relatively easy to extend to more complicated physics, such as porous media [11–13], or multiphase flows [14, 15]. The use of the LBM for topology optimization was pioneered by Pingen *et al.* [16], who used the density approach to topology optimization. The work is extended to multiphase flow problems by Makhija *et al.* [8]. In addition, a number of studies have investigated a level-set based optimization approach using the LBM [17–19].

For gradient-based optimization, a common technique for computing the necessary gradients is the adjoint method. There are two main approaches to this technique, the “discretize then optimize” approach, in which the adjoint problem is derived from the discrete lattice Boltzmann equation, and the “optimize then discretize” approach, in which the adjoint problem is derived from the continuous Boltzmann equation and then discretized. In this paper, we adopt the first approach. The discretize then optimize approach was first used by Tekitek *et al.* [20] for finding optimal parameters for a lattice Boltzmann model, while Krause *et al.* [21] applied the discretize then optimize approach to flow control problems.

Regardless of the computational method used to obtain flow profiles, the vast majority of studies on fluid topology optimization have only considered steady-state flow. On the other hand, many fluidic systems of interest are dominated by unsteady flows and do not permit steady-state solutions. Only few studies have been published treating such systems, however. An implementation of unsteady flow topology optimization using a discrete formulation of the objective function has been presented by Kreissl *et al.* [22], while Deng *et al.* presented a continuous formulation [23]. Both of these studies computed the flow profiles by a finite element discretization of the NS equations. While Yonekura and Kanno [24] have presented a method for computing steady state designs using transient information, to the authors’ knowledge, there are no published works which apply the LBM to topology optimization of unsteady flow problems. In addition, the above works consider problems in which unsteady effects have little influence on the topology of the computed solutions. In the present work, problems with inherent unsteady characteristics will be presented. The problems considered are restricted to two spatial dimensions, but the method is readily extensible to three-dimensional problems.

Following the work of Kreissl *et al.* [22], the unsteady flow topology optimization problems considered in this paper may be written in the time-discrete form:

$$\begin{aligned} \min_{\mathbf{s}} Z(\mathbf{f}^0, \dots, \mathbf{f}^{N_t}, \mathbf{s}) &= \chi \left(\sum_{n=0}^{N_t} z^n(t^n, \mathbf{f}^n, \mathbf{s}^{\text{ph}}) \right), \\ \text{s.t. } \begin{cases} \mathbf{s}, & \text{satisfies design constraints } G_j \leq 0, \\ \mathbf{f}^n, & \text{satisfies the governing equations, } \mathbf{R}^n = \mathbf{0}, \\ & \text{for the given } \mathbf{s}, \forall n \in \{0, \dots, N_t\}, \end{cases} \end{aligned} \quad (1)$$

where \mathbf{s} is a vector of design variables, and \mathbf{s}^{ph} is a vector of physical variables associated with the fluid medium at each design element. The physical variables are obtained from \mathbf{s} by a continuous mapping $X : \mathbf{s} \rightarrow \mathbf{s}^{\text{ph}}$. The vector \mathbf{f}^n is the fluid state at time step $n \in \{0, \dots, N_t\}$, and $\mathbf{R}^n = \mathbf{0}$ is the residual vector of the governing equations at time step n . The objective Z is given by a differentiable function χ which depends on a sum of contributions z^n in each time step. The functions G_j represent the design constraints.

The rest of the paper is organized as follows: section 2 contains a brief overview of the governing equations of the lattice Boltzmann method, including the treatment of boundary conditions and modelling of fluid and solid domains. Section 3 covers the adjoint sensitivity analysis for the case of an unsteady objective function. In section 4 a brief overview of filtering techniques by means of a proper choice of the mapping X is given. The introduced concepts are demonstrated in section 5 with two numerical examples. Finally, section 6 offers a summary of the results as well as concluding remarks.

2. Governing equations

Unlike the Navier-Stokes equations, the LBM models the motion of a fluid as an ensemble of microscopic particles. The state of the system is given in terms of a distribution function, from which macroscopic quantities such as density and fluid velocity can be obtained by computing its moments.

2.1. The lattice Boltzmann method

As shown by He and Luo [25, 26], the discrete lattice Boltzmann equation with Bhatnagar-Gross-Krook (BGK) collision operator [27] reads:

$$f_\alpha(\mathbf{x}_i + \mathbf{c}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}_i, t) - \frac{1}{\tau} [f_\alpha(\mathbf{x}_i, t) - f_\alpha^{\text{eq}}(\mathbf{x}_i, t)], \quad (2)$$

where τ is the relaxation time, \mathbf{c}_α belongs to some discrete set of lattice velocities, and $f_\alpha = f(\mathbf{x}_i, \mathbf{c}_\alpha, t)$ and $f_\alpha^{\text{eq}} = f^{\text{eq}}(\mathbf{x}_i, \mathbf{c}_\alpha, t)$ is respectively the distribution function and local equilibrium associated with the corresponding velocity. For this work, the common D2Q9 scheme (figure 1), which partitions two-dimensional velocity space into nine discrete velocities, has been applied. The discrete velocities are given by:

$$\begin{aligned} & [\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5, \mathbf{c}_6, \mathbf{c}_7, \mathbf{c}_8] \\ &= \frac{\Delta x}{\Delta t} \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}. \end{aligned} \quad (3)$$

For simplicity, the step sizes Δx and Δt may be taken to be in “lattice units”, meaning that $\Delta x = \Delta t = 1$.

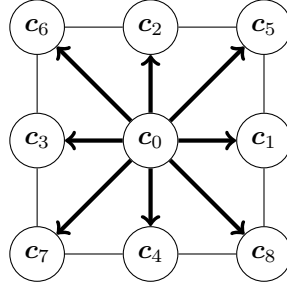


Figure 1: The D2Q9 model.

Typically, equation (2) is separated into a local collision step

$$\tilde{f}_\alpha(\mathbf{x}_i, t) = f_\alpha(\mathbf{x}_i, t) - \frac{1}{\tau} [f_\alpha(\mathbf{x}_i, t) - f_\alpha^{\text{eq}}(\mathbf{x}_i, t)], \quad (4a)$$

followed by a global propagation step

$$f_\alpha(\mathbf{x}_i + \mathbf{c}_\alpha \Delta t, t + \Delta t) = \tilde{f}_\alpha(\mathbf{x}_i, t). \quad (4b)$$

From equation (4) the corresponding residual vectors may be written as

$$\begin{aligned} \mathbf{R}^0 &= \mathbf{f}^0 - \mathbf{f}^{\text{init}}, \\ \mathbf{R}^n &= \mathbf{f}^n - S[\mathbf{f}^{n-1} + \frac{1}{\tau}(\mathbf{f}^{n-1} - \mathbf{f}^{\text{eq}, n-1})], \end{aligned} \quad (5)$$

where S is the streaming operator, which shifts the distribution values as in in (4b), and \mathbf{f}^{init} is the vector of initial distribution values.

In this work we apply the incompressible lattice Boltzmann method introduced by He and Luo [28]. In this model the equilibrium distribution f^{eq} is given by the following second order expansion of the Maxwell-Boltzmann distribution:

$$f_\alpha^{\text{eq}}(\mathbf{x}_i, t) = w_\alpha \left(\rho + \rho_0 \left[3(\mathbf{c}_\alpha \cdot \mathbf{u}) + \frac{9}{2}(\mathbf{c}_\alpha \cdot \mathbf{u})^2 - \frac{3}{2}\mathbf{u}^2 \right] \right). \quad (6)$$

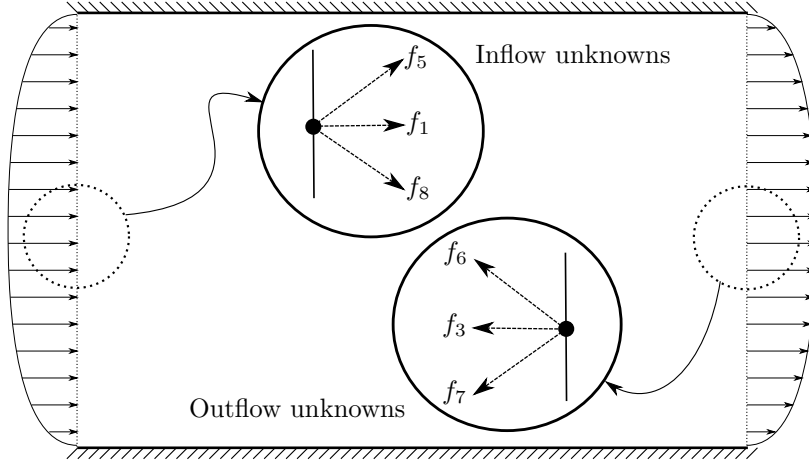


Figure 2: Illustration of boundary conditions in a rectangular domain. At the boundary nodes, the incoming distributions need to be determined by some scheme depending on the type of boundary.

where ρ is the macroscopic density, which fluctuates slightly around the constant value ρ_0 , \mathbf{u} is the macroscopic velocity, and w_α are weights that depend on the discretization in velocity space. For the D2Q9 lattice, the weights are given by

$$w_\alpha = \begin{cases} 4/9, & \text{for } \alpha = 0, \\ 1/9, & \text{for } \alpha \in \{1, 2, 3, 4\}, \\ 1/36, & \text{for } \alpha \in \{5, 6, 7, 8\}. \end{cases} \quad (7)$$

Without loss of generality, the constant term ρ_0 may be taken to be unity. The macroscopic parameters, i.e. density, velocity, pressure, and viscosity can then be evaluated as:

$$\rho(\mathbf{x}_i, t) = \sum_{\alpha=0}^8 f_\alpha(\mathbf{x}_i, t), \quad (8a)$$

$$\rho_0 \mathbf{u}(\mathbf{x}_i, t) = \sum_{\alpha=0}^8 \mathbf{c}_\alpha f_\alpha(\mathbf{x}_i, t), \quad (8b)$$

$$p(\mathbf{x}_i, t) = c_s^2 \rho(\mathbf{x}_i, t), \quad (8c)$$

$$\nu = \left(\tau - \frac{1}{2} \right) c_s^2 \frac{(\Delta x)^2}{\Delta t}, \quad (8d)$$

where p is the scalar pressure, ν is the kinematic viscosity, and c_s is the lattice speed of sound for the D2Q9 lattice. It can be shown that, in lattice units, $c_s^2 = 1/3$.

Assuming low Mach number flow, it can be shown—e.g. by the Chapman-Enskog expansion [10, 29] or asymptotic analysis [30]—that the macroscopic values obtained by the above numerical scheme solve the incompressible NS equations for an isothermal fluid to first order accuracy in time and second order accuracy in space.

2.2. Boundary conditions

Enforcing boundary conditions in the LBM requires special treatment, since the governing equations describe the dynamics of the distributions functions f_i , but hydrodynamic boundary conditions are typically given in terms of the macroscopic properties of the fluid. For a general introduction to the implementation of boundary conditions in the LBM, the reader is referred to the book by Succi [31]. In this section, only the boundary conditions relevant to the numerical examples shall be discussed.

The basic idea of the implementation is that on the boundaries, there are certain velocity distributions that are unknown after the streaming step, since they can be thought of as having entered the computational domain from “outside”. Which distribution values are unknown depends on the orientation of the boundary, as illustrated in figure 2.

For inflow boundaries, a velocity inlet condition is imposed, using the method described by Zou and He [32]. Assuming a bounceback condition in the non-equilibrium part of the velocity distributions normal to the boundary (e.g. $f_1 - f_1^{\text{eq}} = f_3 - f_3^{\text{eq}}$ for an inflow on the west boundary as in figure 2), a closed system of equations is obtained. In the case of an inflow on the west boundary, solving this system for the unknowns yields

$$f_1 = f_3 + \frac{2}{3}\rho_0\bar{u}_x, \quad (9a)$$

$$f_5 = \frac{1}{6}\rho_0\bar{u}_x + \frac{1}{2}\rho_0\bar{u}_y + \frac{1}{2}(f_4 - f_2) + f_7, \quad (9b)$$

$$f_8 = \frac{1}{6}\rho_0\bar{u}_x - \frac{1}{2}\rho_0\bar{u}_y + \frac{1}{2}(f_2 - f_4) + f_6, \quad (9c)$$

where \bar{u}_x and \bar{u}_y is the prescribed velocity in the x - and y -direction, respectively. Similarly structured equations may be derived for other boundary orientations. These boundary conditions are also applied for no-slip closed boundaries, by simply setting $\bar{u}_x = \bar{u}_y = 0$ in (9).

For open boundaries, the zero normal shear stress (ZNS) boundary condition, also known as the no-friction condition, is imposed. This is a Neumann type boundary condition, and a method for implementing it in the LBM has been derived by Junk and Yang using asymptotic analysis [33]. Their derivation gives a more general scheme for the computation of boundary values, but here the equations shall simply be given in the form relevant to this paper, i.e. two dimensional flow with BGK collision operator. For an outflow located on the east boundary (figure 2), the unknown distributions may be computed as:

$$f_3(\mathbf{x}, t) = F_3^{\text{eq}}(1, \mathbf{u}(\mathbf{x}, t - \Delta t)) - (2\nu\tau^{-1} - 1) \times (f_1(\mathbf{x}, t - \Delta t) - f_1^{\text{eq}}(\mathbf{x}, t - \Delta t)), \quad (10a)$$

$$f_6(\mathbf{x}, t) = F_6^{\text{eq}}(1, \mathbf{u}(\mathbf{x}, t - \Delta t)) - 2\nu\tau^{-1}(w_6/w_3) \times (f_1(\mathbf{x}, t - \Delta t) - f_1^{\text{eq}}(\mathbf{x}, t - \Delta t)), \quad (10b)$$

$$f_7(\mathbf{x}, t) = F_7^{\text{eq}}(1, \mathbf{u}(\mathbf{x}, t - \Delta t)) - 2\nu\tau^{-1}(w_7/w_3) \times (f_1(\mathbf{x}, t - \Delta t) - f_1^{\text{eq}}(\mathbf{x}, t - \Delta t)), \quad (10c)$$

where

$$F_i^{\text{eq}}(\rho, \mathbf{u}) = w_i \left(\rho + \rho_0 \left[3(\mathbf{c}_i \cdot \mathbf{u}) + \frac{9}{2}(\mathbf{c}_i \cdot \mathbf{u})^2 - \frac{3}{2}\mathbf{u}^2 \right] \right).$$

Again, equations of similar structure can be applied for different boundary orientations.

As a final remark, we note that in general, corner nodes need to be treated as special cases, since there are effectively five unknown distribution values rather than three. In the numerical examples, all corner nodes are no-slip nodes, and the on-grid bounceback scheme as described in [31] is applied. In this scheme the unknown distribution values are simply set equal to the known values in the reverse directions, i.e.:

$$f_1 \leftrightarrow f_3, \quad f_2 \leftrightarrow f_4, \quad f_5 \leftrightarrow f_7, \quad f_6 \leftrightarrow f_8. \quad (11)$$

This simple scheme ensures that the no-slip condition is satisfied provided that the initial distribution values for opposite velocities are equal.

2.3. Modeling fluid and solid domains in the LBM

For the purpose of topology optimization, the above LBM needs to be modified to allow for a continuous transition between fluid and solid nodes in the optimization procedure. One way to achieve this is to modify

the method such that in solid regions, the Brinkman equation for flow in porous media is satisfied:

$$\frac{1}{\rho} \nabla p = -\frac{\nu}{K} \mathbf{u} + \tilde{\nu} \nabla^2 \mathbf{u}, \quad (12)$$

where K is the permeability of the medium, and $\tilde{\nu}$ is the so-called effective viscosity.

One approach which obeys the Brinkman equation in solid nodes is to adopt a partial bounceback formulation of the LBM. Partial bounceback models were first suggested by Dardis and McCloskey [34, 35]; they are probabilistic meso-scale models in which a bounceback like term dependent on the permeability of the lattice node is added to the collision step. In this work, the partial bounceback model presented by Zhu and Ma [13] is adopted. The modified collision step has the form

$$\tilde{f}_\alpha(\mathbf{x}, t) = f_\alpha^c(\mathbf{x}_i, t) + g(s_i^{\text{ph}})[f_{\alpha'}^c(\mathbf{x}_i, t) - f_\alpha^c(\mathbf{x}_i, t)], \quad (13)$$

where

$$f_\alpha^c = f_\alpha(\mathbf{x}_i, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}_i, t) - f_\alpha^{\text{eq}}(\mathbf{x}_i, t)]$$

is the original post-collision value, $f_{\alpha'}^c$ is the post-collision value in the direction opposite f_α , as per (11), and $g \in [0, 0.5]$ is the fraction of particles bounced back at the node i . One can then choose $g : [0, 1] \rightarrow [0, 0.5]$ to be a continuous function such that

$$g(0) = 0.5, \quad g(1) = 0,$$

with $s_i^{\text{ph}} = 1$ corresponding to a fluid node, which recovers the standard collision step, and $s_i^{\text{ph}} = 0$ corresponding to a solid node, where a bounceback like collision step is obtained. Following Borrvall and Petersson [3], the following convex interpolation function is chosen:

$$g(s_i^{\text{ph}}) = 0.5 \left(1 - s_i^{\text{ph}} \frac{1 + \gamma}{s_i^{\text{ph}} + \gamma} \right), \quad (14)$$

where γ is an adjustable parameter which allows controlled penalization of intermediate permeabilities in the optimized designs. Increasing the value of γ results in a more linear scaling, which causes a greater penalization of intermediate values, see figure 3. While the function (14) was formulated by Borrvall and Petersson specifically for minimal dissipation type problems, it has also been found to work well for the numerical examples given in the present work.

3. Adjoint sensitivity analysis

A key component of the topology optimization procedure is the efficient evaluation of the gradients of the objective function and constraints with respect to the design variables. A method commonly used in topology optimization for determining the gradients is the adjoint method, which allows for all of the sensitivities to be computed from the transposed solution of a single linear system of equations.

Following the derivation by Kreissl *et al.* [22], the time discrete quantities are collected in column vectors:

$$\begin{aligned} \hat{\mathbf{z}} &= [z_0, \dots, z_{N_t}]^T, \\ \hat{\mathbf{R}} &= [\mathbf{R}^0, \dots, \mathbf{R}^{N_t}]^T, \\ \hat{\mathbf{f}} &= [\mathbf{f}^0, \dots, \mathbf{f}^{N_t}]^T, \\ \hat{\boldsymbol{\lambda}} &= [\boldsymbol{\lambda}^0, \dots, \boldsymbol{\lambda}^{N_t}]^T, \end{aligned} \quad (15)$$

where $\boldsymbol{\lambda}^n$, $n \in \{0, \dots, N_t\}$ is the vector of Lagrange multipliers at time step n . The adjoint equation for objective functionals of the form (1) is then given by

$$\left(\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{f}}} \right)^T \hat{\boldsymbol{\lambda}} = - \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{f}}} \right)^T \frac{\partial \chi}{\partial \hat{\mathbf{z}}}. \quad (16)$$

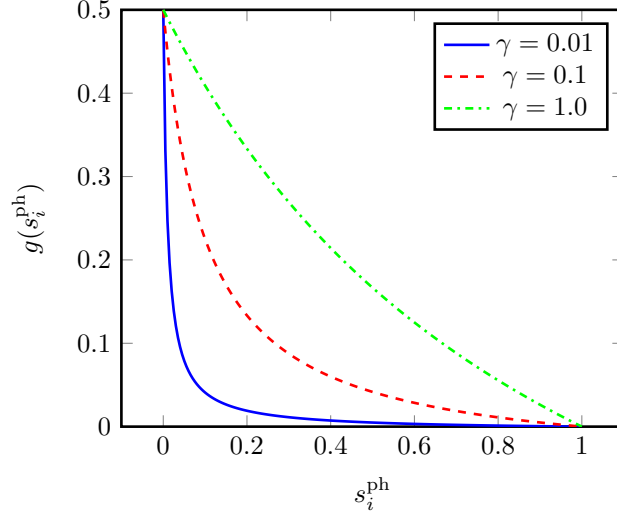


Figure 3: Plot of the interpolation function (14) for different values of γ . Higher values result in more linear scaling and thus penalize “grey” design variables more.

Since the LBM is an explicit time marching scheme dependent only on the state in the previous time step, the Jacobian $\partial \hat{\mathbf{R}} / \partial \hat{\mathbf{f}}$ is a band matrix. For the initial step \mathbf{R}^0 we have

$$\frac{\partial \mathbf{R}^0}{\partial \mathbf{f}^j} = \begin{cases} \mathbf{I}, & \text{for } j = 0, \\ \mathbf{0}, & \text{for } j \neq 0, \end{cases} \quad (17)$$

where \mathbf{I} is the identity matrix. For subsequent time steps, $n > 0$, the following notation is introduced:

$$\frac{\partial \mathbf{R}^n}{\partial \mathbf{f}^j} = \begin{cases} \mathbf{C}^n, & \text{for } j = n - 1, \\ \mathbf{D}^n, & \text{for } j = n, \\ \mathbf{0}, & \text{for } j \notin \{n - 1, n\}. \end{cases} \quad (18)$$

Equation (16) may then be written in matrix form by performing the substitutions

$$\left(\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{f}}} \right)^T = \begin{pmatrix} \mathbf{I} & (\mathbf{C}^1)^T & & & \\ & (\mathbf{D}^1)^T & (\mathbf{C}^2)^T & & \\ & & (\mathbf{D}^2)^T & \ddots & \\ & & & \ddots & (\mathbf{C}^{N_t})^T \\ & & & & (\mathbf{D}^{N_t})^T \end{pmatrix}, \quad (19)$$

$$\left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{f}}} \right)^T \frac{\partial \chi}{\partial \hat{\mathbf{z}}} = \begin{pmatrix} (\partial z^0 / \partial \mathbf{f}^0)^T (\partial \chi / \partial z^0) \\ \vdots \\ (\partial z^{N_t} / \partial \mathbf{f}^{N_t})^T (\partial \chi / \partial z^{N_t}) \end{pmatrix}. \quad (20)$$

The band structure of the matrix (19) allows the adjoint problem to be conveniently solved backward in time, starting by solving for λ^{N_t} :

$$\lambda^{N_t} = (\mathbf{D}^{N_t})^{-T} \left(\frac{\partial z^{N_t}}{\partial \mathbf{f}^{N_t}} \right)^T \frac{\partial \chi}{\partial z^{N_t}}. \quad (21a)$$

Subsequent $\boldsymbol{\lambda}^n$ may then be solved by backward substitution:

$$\boldsymbol{\lambda}^n = (\mathbf{D}^n)^{-T} \left[\left(\frac{\partial z^n}{\partial \mathbf{f}^n} \right)^T \frac{\partial \chi}{\partial z^n} - (\mathbf{C}^{n+1})^T \boldsymbol{\lambda}^{n+1} \right]. \quad (21b)$$

Once $\hat{\boldsymbol{\lambda}}$ has been fully determined, the gradient with respect to the physical variable \mathbf{s}^{ph} may be computed by

$$\frac{dZ}{ds_k^{\text{ph}}} = \frac{\partial Z}{\partial s_k^{\text{ph}}} + \left(\frac{\partial Z}{\partial \hat{\mathbf{z}}} \right)^T \frac{\partial \hat{\mathbf{z}}}{\partial s_k^{\text{ph}}} + (\hat{\boldsymbol{\lambda}})^T \frac{\partial \mathbf{R}}{\partial s_k^{\text{ph}}}. \quad (22)$$

While the matrix form of equation (16) neatly illustrates the overall structure of the adjoint problem, implementing the adjoint analysis using matrix routines requires assembling the matrices \mathbf{D}^n and \mathbf{C}^n in each time step. The computational cost of such an implementation quickly becomes intractable as the number of time steps considered increases. Alternatively, it is possible to exploit the local nature of the lattice Boltzmann equation to derive explicit expressions for the Lagrange multipliers in each time step, thereby bypassing the need for matrix routines entirely; in this way the adjoint problem can be solved on the same time scale as the forward problem. A detailed derivation of a discrete adjoint lattice Boltzmann method has already been given by Liu *et al.* [36], who applied it to steady-state fluid topology optimization problems. Below we give an outline of the derivation of the explicit adjoint expressions, including the treatment of the boundary conditions described in section 2.2, which to our knowledge have not been covered before.

3.1. Derivation of adjoint equations

For the purpose of this derivation, we will consider an arbitrary objective of the form (1). From (21) it is apparent that there will be a contribution from the source term $(\partial z^n / \partial \mathbf{f}^n)(\partial \chi / \partial z^n)$ for each multiplier $\lambda_\alpha(\mathbf{x}, t)$. We will denote this contribution

$$G_\alpha(\mathbf{x}, t) = - \left(\frac{\partial z^t}{\partial f_\alpha(\mathbf{x}, t)} \right)^T \frac{\partial \chi}{\partial z^t}. \quad (23)$$

Beyond this ubiquitous contribution, it is necessary to derive different equations for interior nodes and each type of boundary node. The various types will be considered in turn.

3.1.1. Interior nodes

Since the interior nodes only require information about distribution values in the previous time step, the factor $(\mathbf{D}^n)^{-T}$ in (21b) can effectively be considered to be an identity term for the purpose of the interior nodes. What is left to consider is then the term $(\mathbf{C}^{n+1})^T \boldsymbol{\lambda}^{n+1} = (\partial \mathbf{R}^{n+1} / \partial \mathbf{f}^n)^T \boldsymbol{\lambda}^{n+1}$. The non-zeros terms for a given node will be at the neighboring residuals, i.e. the nodes where information has been propagated to. Specifically, the individual Lagrange multipliers may be obtained as

$$\lambda_\alpha(\mathbf{x}, t) = G_\alpha(\mathbf{x}, t) - \sum_{j=0}^8 \frac{\partial R_j(\mathbf{x} + \mathbf{c}_j \Delta t, t + \Delta t)}{\partial f_\alpha(\mathbf{x}, t)} \lambda_j(\mathbf{x} + \mathbf{c}_j \Delta t, t + \Delta t), \quad (24)$$

where $\alpha \in \{0, \dots, 8\}$ corresponds to a multiplier for each distribution value at the node. Because of the non-linearity of the collision step, the sum of differentials in (24) results in a quite formidable expression. A practical way to implement this is to automatically generate the code for evaluating it using a computer algebra system, i.e. Maple.

3.1.2. Inflow boundaries

We will consider an inflow on the western boundary of the domain, such that the unknown distributions are computed by (9). The necessary adjoint computations on this boundary differ from those of the interior nodes in one significant way: since three distributions are computed using distribution values from the same

time step, the matrix factor $(D^n)^{-T}$ can no longer be considered an identity. This can be handled by multiplying the set of nine Lagrange multipliers in each inflow point by a 9×9 matrix. For a west inflow, the matrix is

$$A = \left(\frac{\partial R_j(\mathbf{x}, t)}{\partial \mathbf{f}(\mathbf{x}, t)} \right)^{-T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (25)$$

that is, the inverse transpose of the residual Jacobian at the point \mathbf{x} . The nine multipliers at each inflow node are then found by computing (24) and multiplying the result with the matrix (25). For the bounceback condition on the corners, the Lagrange multipliers may be computed similarly.

3.1.3. Open boundaries

For the open boundaries, the unknown distributions are computed using only information from the previous time step, just like the interior nodes. Thus, a matrix like (25) is not needed. Rather, equation (24) needs a slight adjustment:

$$\begin{aligned} \lambda_\alpha(\mathbf{x}, t) &= G_\alpha(\mathbf{x}, t) \\ &\quad - \sum_{j \in \Lambda_s} \frac{\partial R_j(\mathbf{x} + \mathbf{c}_j \Delta t, t + \Delta t)}{\partial f_\alpha(\mathbf{x}, t)} \lambda_j(\mathbf{x} + \mathbf{c}_j \Delta t, t + \Delta t) \\ &\quad - \sum_{j \in \Lambda_u} \frac{\partial R_j(\mathbf{x}, t + \Delta t)}{\partial f_\alpha(\mathbf{x}, t)} \lambda_j(\mathbf{x}, t + \Delta t), \end{aligned} \quad (26)$$

here Λ_s refers to the set of velocities that can stream to a neighbour within the domain, and Λ_u is the set of unknown distributions. For an open boundary on the east part of the domain, $\Lambda_s = \{0, 2, 3, 4, 6, 7\}$, and $\Lambda_u = \{3, 6, 7\}$.

3.2. Implementation

The resulting algorithm for obtaining the objective and sensitivities is summarized by pseudo code in algorithm 1. Note that equation (24) results in expressions which depend on $\mathbf{f}(\mathbf{x}, t)$, meaning that the backward time stepping in the adjoint problem requires the value of the state variable \mathbf{f}^n in every time step $n \in \{N_t, N_t - 1, \dots\}$. In this work, we have taken the naive approach of simply storing the full history in memory. For larger scale problems (i.e. in 3D), storing the entire history of the forward analysis would be prohibitively memory intensive. However, the memory requirement can be significantly reduced—at the cost of extra computation—by implementing a checkpointing algorithm, c.f. [37, 38].

4. Filtering

For topology optimization problems in unsteady fluid dynamics, taking the physical and design variables to be equal, i.e. $\mathbf{s} = \mathbf{s}^{\text{ph}}$, can lead to numerical problems in the optimization procedure. Since the design and physical variables are equivalent, the optimizer can add features to the design on the length scale of a single grid point. This makes it very difficult to obtain reasonable solutions, because the large search space combined with the objective functions non-linear response to the design variables makes it unlikely that the optimization procedure will converge to a well defined design. Rather, designs will typically converge to local minima with poorly resolved topologies. This pathology arises in both the numerical examples given below. To alleviate such issues, several regularization techniques have been proposed in the literature [1].

Algorithm 1 Compute Z and dZ/ds

```
for all  $n \in \{1 \dots N_t\}$  do
  Advance current time step from  $n - 1$  to  $n$  by the lattice Boltzmann equation (2).
  Save the vector  $\mathbf{f}^n$  in memory.
end for
Compute the objective  $Z$  by equation (1).
for all  $n \in \{N_t \dots 1\}$  do
  if  $n = N_t$  then
    Compute  $\lambda^{N_t}$  by (21a).
  else
    Advance current time step from  $n$  to  $n - 1$  by equation (24).
  end if
end for
Compute  $dZ/ds^{\text{ph}}$  by equation (22).
Compute  $dZ/ds$  from  $dZ/ds^{\text{ph}}$  by the chain rule (see section 4).
```

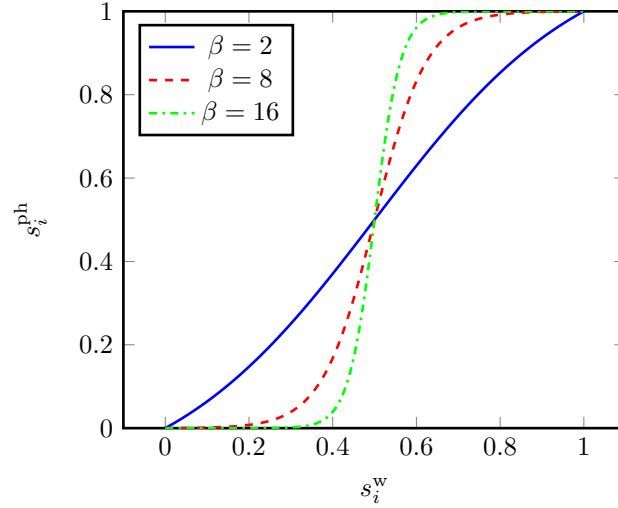


Figure 4: Plot of the projection function (27) with a threshold of $\eta = 0.5$ for different values of β .

For this work, in order to regularize the optimized design and obtain a discrete 0/1 result, a Heaviside projection filter is applied. Several variation of this idea have been described in published works [39–41]. The steps necessary for applying the projection filter are briefly described below.

4.1. Threshold projection

In order to obtain a black and white solution, the following smoothed Heaviside function given by Wang *et al.* [42] is applied:

$$s_i^{\text{ph}} = \frac{\tanh(\beta\eta) + \tanh(\beta(s_i^w - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}. \quad (27)$$

Here, β is a control parameter for the function which determines the steepness of the region around the threshold, see figure 4, and η gives the threshold for the projection, so that values above η are projected toward 1 and values below are projected toward 0. Note that in the limit $\beta \rightarrow 0$, $s_j^w = s_j^{\text{ph}}$, so that the unprojected variable is recovered. The variable s_j^w refers to a smoothing of the design variables given by the

weighted averaging

$$s_j^w = \frac{\sum_{k \in \mathbb{N}_{e,j}} H(\mathbf{x}_k) s_k}{\sum_{k \in \mathbb{N}_{e,j}} H(\mathbf{x}_k)}, \quad (28)$$

where $\mathbb{N}_{e,j}$ is the neighborhood of elements within a filter radius r , and $H(\mathbf{x}_k)$ is the weighting factor given by

$$H(\mathbf{x}_k) = r - |\mathbf{x}_k - \mathbf{x}_j|, \quad (29)$$

where \mathbf{x}_j and \mathbf{x}_k are the center point coordinates of the design elements j and k .

Once the design variables have been mapped to the physical variables by applying the projection technique, the sensitivities of the objective function with respect to the original design variables can be computed using the chain rule:

$$\frac{\partial Z}{\partial s_k} = \sum_{j \in \mathbb{N}_{e,k}} \frac{\partial Z}{\partial s_j^{\text{ph}}} \frac{\partial s_j^{\text{ph}}}{\partial s_j^w} \frac{\partial s_j^w}{\partial s_k}, \quad (30)$$

where

$$\frac{\partial s_j^w}{\partial s_k} = \frac{H(\mathbf{x}_k)}{\sum_{k \in \mathbb{N}_{e,j}} H(\mathbf{x}_k)}. \quad (31)$$

We emphasize that the projection filter does not necessarily ensure mesh independence in the final design [42]. This can be alleviated by employing the robust formulation given by Wang *et al.* [42], or by geometric constraints as presented by Zhou *et al.* [43], but application of these techniques to fluid topology optimization has yet to be investigated.

4.2. β -continuation

As shown in figure 4, increasing the value of the parameter β has the result that the threshold function (27) will project a greater range of values towards a pure solid or fluid node. Therefore, ideally the value of β should be large in order to ensure a crisp black and white design. However, using an initially large value of β typically leads to numerical problems such as design oscillations, since the sensitivities for design variables near the threshold η will be comparatively large. Instead, a common strategy is the β -continuation approach. The optimization is run with an initially small value of β ; the parameter is then periodically increased to facilitate proper convergence to a black and white design. This strategy has been employed for both the numerical examples given below.

5. Numerical examples

The topology optimization framework is now exemplified by two numerical examples. In both cases, a domain is considered where the fluid is initially at rest, and all distribution functions are at their equilibrium values.

The design is updated in each iteration by the method of moving asymptotes (MMA), due to Svanberg [44]. The domain is initialized to some initial design guess (which will be detailed below), and the optimization procedure is considered to be converged if

$$\|\mathbf{s}^m - \mathbf{s}^{m-1}\|_\infty < \epsilon, \quad (32)$$

where m is the current iteration step, and ϵ is some small tolerance parameter. For the examples below, the value $\epsilon = 10^{-4}$ has been used.

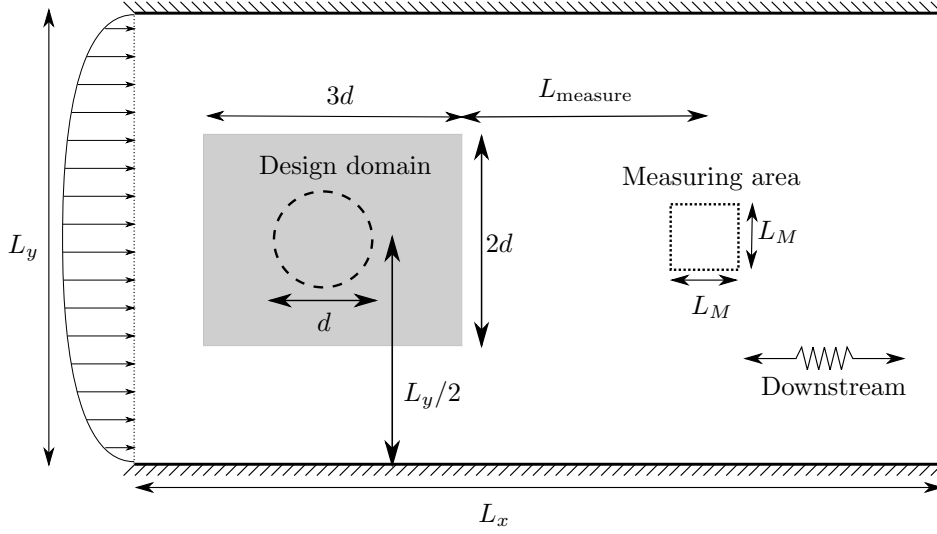


Figure 5: Illustration of the computational domain for the flow control problem. Note that the downstream section of the computational domain is much longer than the figure would indicate, but this is irrelevant for illustrative purposes.

5.1. Controlling flow past an obstacle

For the first numerical example, we consider a problem that exhibits significant unsteady effects, but in which an exact (but not necessarily unique) global optimum is known to exist. It is well known that even at relatively moderate Reynolds numbers, in the wake of an obstacle, complicated flow patterns can arise. The goal of this problem is to control the topology of an obstacle in order to recover a given spatial and temporal flow profile. The geometry of the problem is shown in figure 5. To obtain a reference flow profile, an initial simulation is run with a cylindrical obstacle (as shown with a dashed contour in figure 5). The flow is allowed to develop for a set number of time steps to ensure that vortex shedding has started. For the remaining time steps, the temporal profile of the velocities in the measuring area is then saved. During the optimization procedure, we seek to minimize the time-averaged difference between the reference velocity profile and the profile of the obstacle generated by topology optimization.

Mathematically, the optimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{s}} Z &= \log \sum_{n=N_0}^{N_t} \sum_{h \in M} \frac{\|\mathbf{u}(\mathbf{x}_h, t_n) - \mathbf{u}_{\text{ref}}(\mathbf{x}_h, t_n)\|^2}{N_m M_A}, \\ \text{s.t.} \quad &\begin{cases} G_0 = \frac{1}{N_s} \sum_{k=1}^{N_s} s_k^{\text{ph}} - 0.9 \leq 0, \\ G_1 = \Delta p - \xi \Delta p_{\text{ref}} \leq 0, \\ 0 \leq s_k \leq 1, \quad \forall k = 1, \dots, N_s. \end{cases} \end{aligned} \quad (33)$$

Here, M denotes the nodes in the measuring domain, and M_A denotes its area. The deviation from the reference velocity is computed over $N_m = N_t - N_0$ time steps. The problem is constrained by the volume constraint G_0 which constrains the amount of fluid allowed in the design domain, corresponding to the fluid amount around the reference obstacle. Additionally, the constraint G_1 is a pressure drop constraint, with Δp being the time averaged pressure drop in the cavity. The reference pressure drop Δp_{ref} is the pressure drop in the cavity for the cylinder, and ξ is a constant tolerance parameter.

Because the spatial and temporal flow profile is dependent on the obstacle in a highly non-linear way, design features on length scales down to one node can have significant effects on the flow in the wake of the obstacle. Therefore, in order to regularize the problem, filtering is applied.

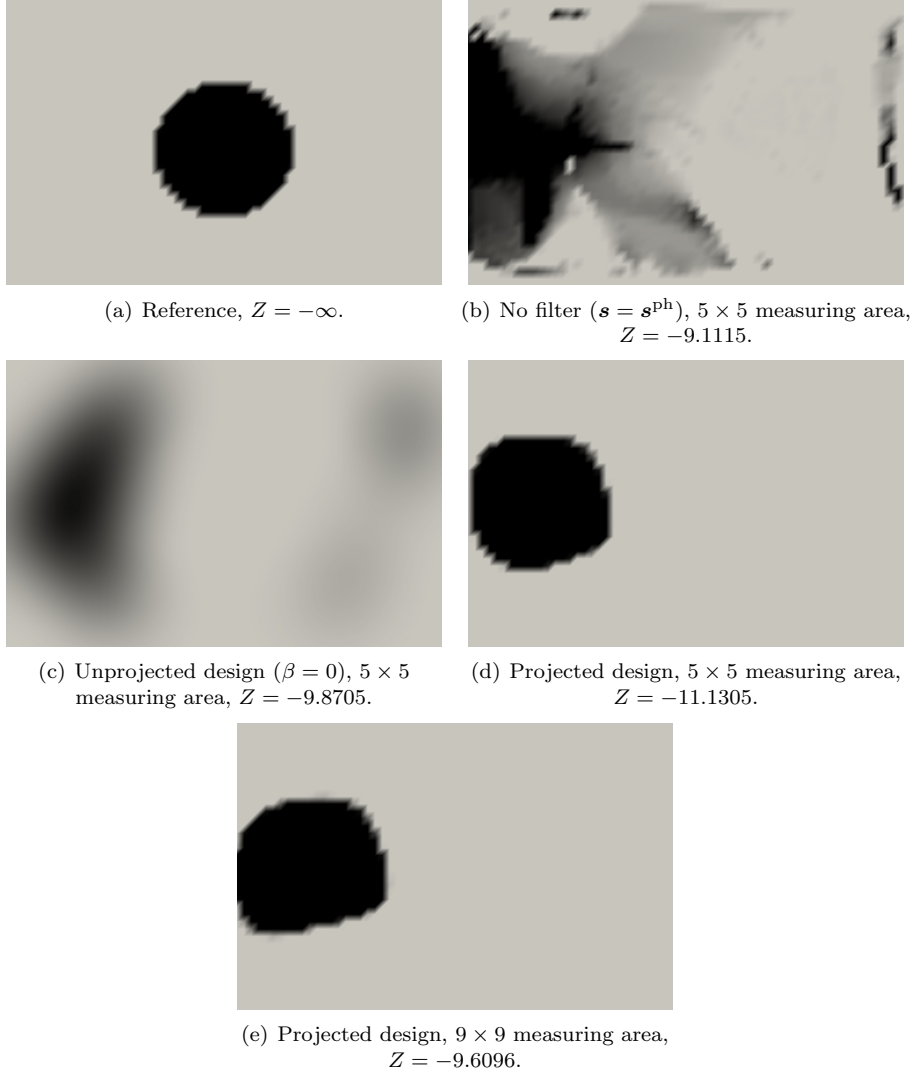


Figure 6: Example designs for the flow control problem. Only the design domain is shown.

To ensure convergence to a black and white design, the β -continuation approach described in section 4.2 is employed. The initial value of $\beta = 1$ is used, which is increased by the constant factor 1.4 every 50 iterations until a maximal value $\beta_{\max} = (1.4)^{15}$ is reached. For the projection, the value $\eta = 0.5$ has been used. Note that in order to ensure the existence of a globally optimal solution, the reference velocity is obtained by performing projection on the initial cylinder, with the value of β set to its maximum value, and then measuring the resulting flow profile.

As the initial design, the design domain is initialized to be equal to the volume fraction everywhere. For the purpose of the filter, the vertical length L_y will be considered scaled to unity, so that a filter radius $0 < r < 1$ may be chosen independently of the mesh refinement. The remaining numerical values used for the problem are listed in table 1, and example obstacle designs are shown in figure 6. For these examples, the Reynolds number with respect to the initial cylinder diameter is 60.

From the examples in figure 6(b)–6(c), it is clear that the projection filter as well as the β -continuation strategy is essential to ensure convergence to a well defined black and white solution. In all cases, the known global optimum is not found, but with projection a reasonable obstacle is obtained for different sizes of the

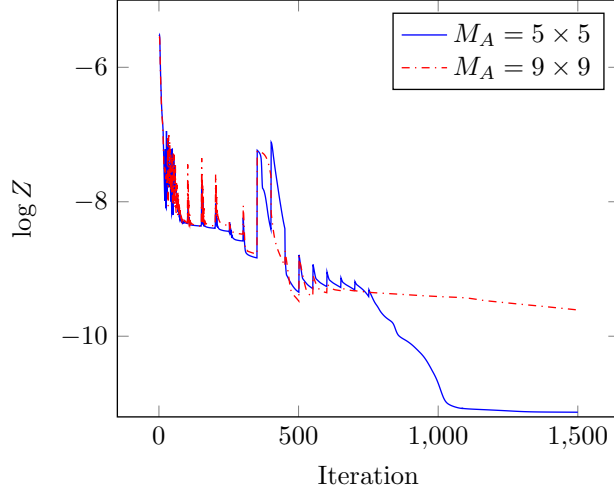


Figure 7: Convergence plot of the objective for the examples in figure 6(d) and 6(e).

L_x	400
L_y	100
L_{measure}	$L_x/4$
d	$L_y/5$
N_0	10000
N_t	15000
γ	1.0
ξ	1.5
r	0.1
β_{max}	$(1.4)^{15}$

Table 1: Numerical parameters used for the flow control example.

measuring area.

The convergence history for two of the examples is given in figure 7. The spikes in the plot are due to the β -continuation, but it is apparent that these oscillations become smaller and smaller as the optimization progresses. For a larger measuring area we observe that the final objective has a higher value, since increasing the amount of points in which the velocities should match makes the optimization problem harder. That is, increasing the measurement area results in a mathematically stronger objective which is more difficult to minimize. It is not immediately clear which one of the two examples provides the “best” reproduction of cylindrical vortex shedding, however. We discuss this further below. In both cases, the pressure drop constraint is initially violated, but this is handled within 10 iterations of the MMA algorithm, after which both constraints are fulfilled for the remainder of the optimization procedure.

Note that for both examples in figure 6(d)–6(e), the obstacle is shifted to the left compared to the reference obstacle. As can be seen in figure 8, the phases of the vortex shedding match up quite well in spite of this, even though the magnitudes differ. As is apparent from figure 9, the more elongated shape of the computed obstacle results in a longer “tail” in the immediate wake of the obstacle, so that the vortices still match up fairly well in spite of the computed obstacle being shifted to the left.

To conclude this section, some additional comments should be made about the formulation (33). If we consider the two examples in figure 6(d) and 6(e), it is clear that the relative difference between their shape is quite small, and yet the difference in terms of the objective function is substantial. Indeed, it turns out that if the objective of the design optimized for a 5×5 measuring area is computed using a 9×9 measuring area, this design actually performs better than the design specifically optimized for the 9×9 area. Given

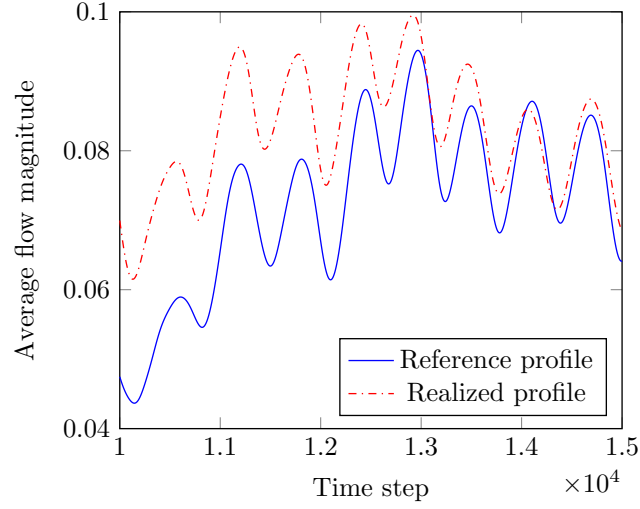
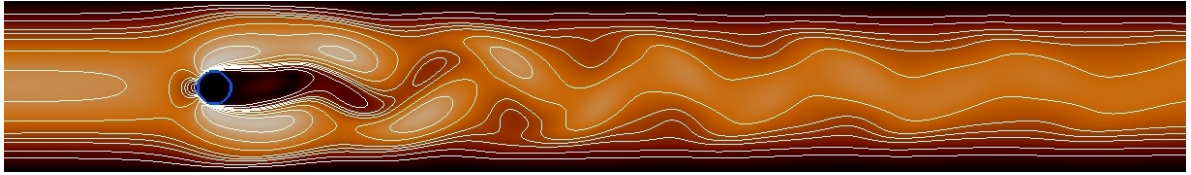
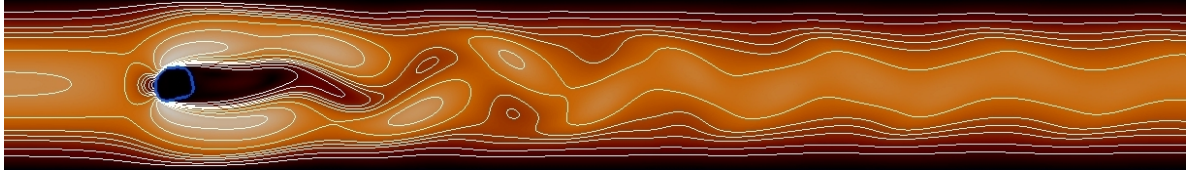


Figure 8: Comparison of the reference and attained profile for the obstacle shown in figure 6(e).



(a) Reference profile.



(b) Obtained profile.

Figure 9: Qualitative comparison of the velocity magnitude contours of the reference design (figure 6(a)) and a computed obstacle (figure 6(e)) in a single time step. The blue contour line indicates the obstacle.

the relatively small difference in the two designs, this demonstrates the non-linear nature of the problem and the likelihood that the optimization algorithm will end up in a local minimum. However, as figure 8 and 9 demonstrate, the performance of such minima can still be quite good.

5.2. A fluid pump

For the second example, we consider the problem of optimizing a simplified fluid pump. The basic idea is to have a vertical inflow on which a periodic inflow is imposed, and then optimize the interior domain to ensure that fluid is transported horizontally from the left inlet of the domain to the right outlet. More precisely, a sinusoidally varying parabolic inflow with maximal velocity u_{\max} given by

$$u_y(t^n) = u_{\max} \sin(\pi t^n / v), \quad (34)$$

is imposed, where v is some frequency. The computational domain is illustrated in figure 10.

The goal is to maximize the horizontal velocity at the open boundary marked “pump outflow”. The second open boundary will effectively act as a reservoir from which additional fluid can enter the domain.

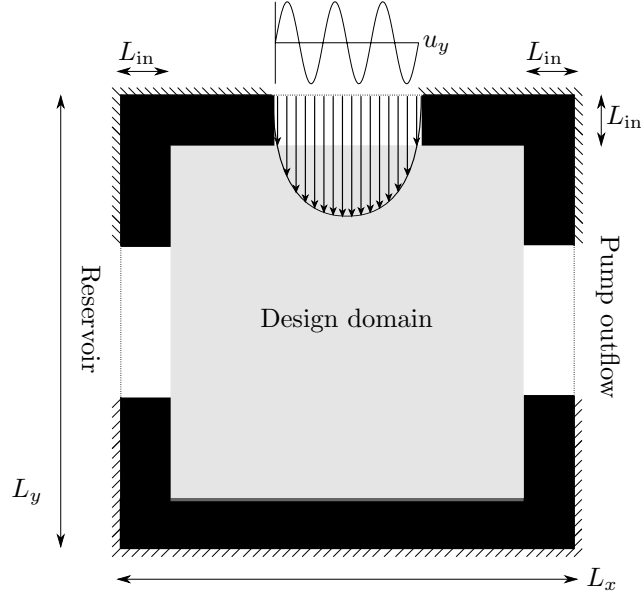


Figure 10: Illustration of the computational domain for the fluid pump problem.

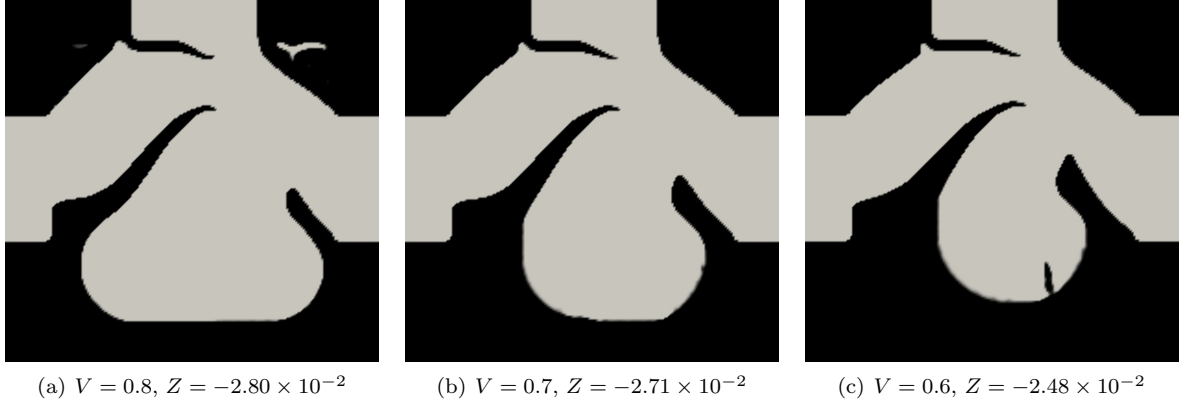


Figure 11: Example pump designs for different values of the volume fraction.

The optimization problem is formulated as follows:

$$\begin{aligned}
 \min_{\mathbf{s}} Z &= -\frac{1}{N_t L_{\text{out}}} \sum_{n=0}^{N_t} \sum_{\mathbf{x}_i \in \Gamma_{\text{out}}} u_x(\mathbf{x}_i, t_n) \\
 \text{s.t.} \quad &\begin{cases} G_0 = \frac{1}{N_s} \sum_{k=1}^{N_s} s_k^{\text{ph}} - V \leq 0, \\ 0 \leq s_k \leq 1, \quad \forall k = 1, \dots, N_s. \end{cases}
 \end{aligned} \tag{35}$$

Here, Γ_{out} denoted the set of grid points at the pump outflow boundary, and L_{out} denoted the length of the boundary. Once again, a volume constraint on the amount of allowed fluid in the design domain is applied to the problem.

As with the obstacle design problem, projection is applied in order to regularize the design and ensure a black and white solution. Once again, the value $\eta = 0.5$ is used for the projection function, and the

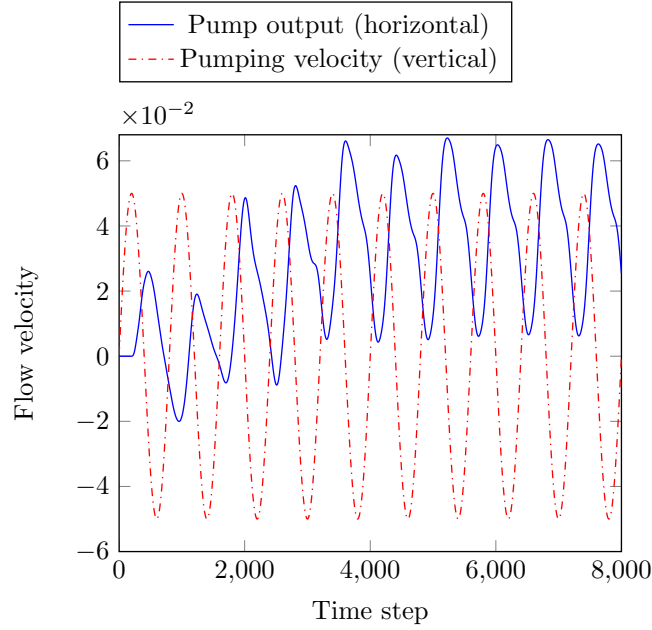


Figure 12: Performance of the pump design in figure 11(a) as a function of time.

value of β is iteratively increased during the course of the optimization procedure. The design variables are initialized to be equal to the volume fraction everywhere. The remaining numerical values for this problem are listed in table 2. The Reynolds number with respect to the top inflow length is 80.

L_x	204
L_y	204
L_{in}	25
N_t	8000
γ	0.5
r	0.03
v	400
u_{max}	0.05
β_{max}	$(1.4)^{15}$

Table 2: Numerical parameters used for the pump example.

Example designs for this problem for different values of the volume constraint V are shown in figure 11. The imposed flow from the top is cyclical, consisting of an inflow phase in which fluid is flowing into the domain from the top, and an outflow phase in which fluid is sucked out from the top; an optimized design needs to account for both these phases. Each design in figure 11 exhibits the same basic feature of a narrowing of the inflow channel, and a reservoir in the middle which is “shielded” by a protrusion on the left. The performance of the design example in figure 11(a) is illustrated by the figures 12 and 13. Figure 12 shows the pump output as a function of the imposed pumping velocity. Note that after the first few cycles, the output is never negative, meaning that fluid never flows back into the domain at the output, even during the outflow cycle. The mechanism by which this is achieved is shown by streamlines in figure 13. During the inflow cycle, fluid from the top flows towards the pumping outlet, while additional fluid is pulled from the left reservoir (left open boundary) towards the central reservoir. During the outflow phase, fluid from the central reservoir flows toward the pumping outlet, while fluid from the left reservoir flows in a vortex like path around the top inlet; this fluid is then transported towards the pumping outlet during the next

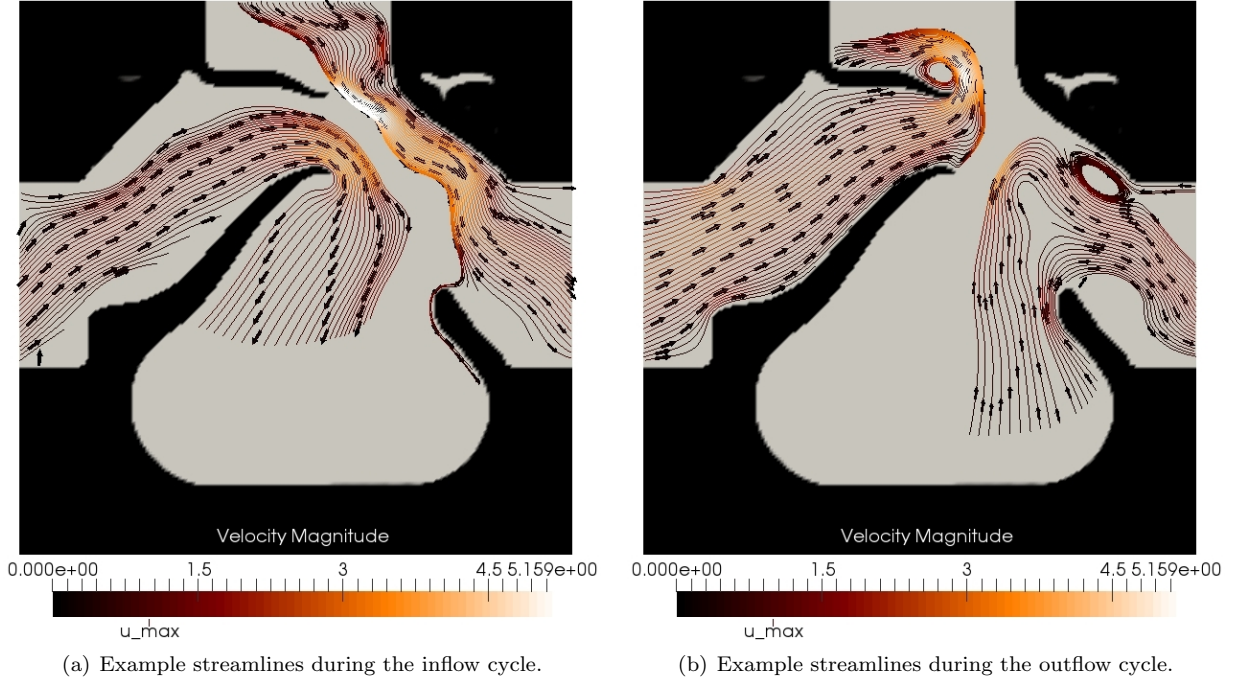


Figure 13: Streamlines of the flow in the design in figure 11(a) during selected timesteps in the inflow and outflow phase. Note that the velocity magnitude has been scaled so that $\|u_{\max}\| = 1$.

inflow phase.

6. Conclusion

Fluid topology optimization using the lattice Boltzmann method has been successfully applied to problems which exhibit significant unsteady effects. The topologies of the computed solutions were described by the density approach, while the fluid-solid boundary conditions were imposed by a partial bounceback model. By solving the lattice Boltzmann equation, and the corresponding discrete adjoint equation, the objective function as well as the design sensitivities were determined in a computationally efficient manner, which allowed solving problems where many time steps were needed to resolve the dynamical effects. In addition, the necessity of a transformation of the design variables in order to ensure convergence to well defined designs has been demonstrated. It is emphasized that the applied map from design to physical variables does not guarantee mesh independence of solutions, but is nonetheless necessary for obtaining well designed designs. While the proposed formulation can be extended to 3D problems, the increase in both computational cost and memory footprint is likely to pose a significant challenge.

Acknowledgements

The authors acknowledge the financial support received from the TopTen project sponsored by the Danish Council for Independent Research, from the NextTop project, sponsored by the Villum foundation, and the Hypercool project sponsored by the Innovation Fund Denmark.

References

- [1] M. P. Bendsøe, O. Sigmund, Topology optimization: theory, methods and applications, Springer, 2004.

- [2] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Computer Methods in Applied Mechanics and Engineering* 71 (2) (1988) 197–224, ISSN 0045-7825.
- [3] T. Borrvall, J. Petersson, Topology optimization of fluids in Stokes flow, *International Journal for Numerical Methods in Fluids* 41 (1) (2003) 77–107, ISSN 14333015, 02683768, 10970363, 02712091, doi:10.1002/fld.426.
- [4] A. Gersborg-Hansen, O. Sigmund, R. B. Haber, Topology optimization of channel flow problems, *Structural and Multidisciplinary Optimization* 30 (3) (2005) 181–192, ISSN 16151488, 1615147x, doi:10.1007/s00158-004-0508-7.
- [5] L. H. Olesen, F. Okkels, H. Bruus, A high-level programming-language implementation of topology optimization applied to steady-state Navier-Stokes flow, *International Journal for Numerical Methods in Engineering* 65 (7) (2006) 975–1001, ISSN 10970207, 00295981, doi:10.1002/nme.1468.
- [6] N. Aage, A. Gersborg-Hansen, O. Sigmund, Topology optimization of large scale Stokes flow problems, *Structural and Multidisciplinary Optimization* 35 (2008) 175–180, ISSN 16151488, 1615147x, doi:10.1007/s00158-007-0128-0.
- [7] C. S. Andreasen, A. R. Gersborg, O. Sigmund, Topology optimization of microfluidic mixers, *International Journal for Numerical Methods in Fluids* 61 (5) (2009) 498–513, ISSN 10970363, 02712091, doi:10.1002/fld.1964.
- [8] D. Makhija, G. Pinggen, R. Yang, K. Maute, Topology optimization of multi-component flows using a multi-relaxation time lattice Boltzmann method, *Computers and Fluids* 67 (2012) 104–114, ISSN 18790747, 00457930, 10960341, 00426822, doi:10.1016/j.compfluid.2012.06.018.
- [9] M. C. Sukop, D. T. Thorne, Jr., *Lattice Boltzmann modelling*, Springer, 2nd edn., 2007.
- [10] D. A. Wolf-Gladrow, *Lattice-gas cellular automata and lattice Boltzmann models - an introduction*, Springer, 2005.
- [11] M. A. A. Spaid, F. Phelan, Lattice Boltzmann methods for modeling microscale flow in fibrous porous media, *Physics of Fluids* 9 (9) (1997) 2468–2474, ISSN 10897666, 10706631, doi:10.1063/1.869392.
- [12] S. D. C. Walsh, H. Burwinkle, M. O. Saar, A new partial-bounceback lattice-Boltzmann method for fluid flow through heterogeneous media, *Computers and Geosciences* 35 (6) (2009) 1186–1193, ISSN 18737803, 00983004, doi:10.1016/j.cageo.2008.05.004.
- [13] J. Zhu, J. Ma, An improved gray lattice Boltzmann model for simulating fluid flow in multi-scale porous media., *Advances in Water Resources* 56 (2013) 61, ISSN 03091708, 18729657.
- [14] T. Inamuro, T. Ogata, S. Tajima, N. Konishi, A lattice Boltzmann method for incompressible two-phase flows with large density differences, *Journal of Computational Physics* 198 (2) (2004) 628–644, ISSN 10902716, 00219991, 10957103, 00219797, doi:10.1016/j.jcp.2004.01.019.
- [15] P. Asinari, Semi-implicit-linearized multiple-relaxation-time formulation of lattice Boltzmann schemes for mixture modeling, *Physical Review E* 73 (5) (2006) 056705, ISSN 15502376, 15393755, doi:10.1103/physreve.73.056705.
- [16] G. Pinggen, A. Evgrafov, K. Maute, Topology optimization of flow domains using the lattice Boltzmann method, *Structural and Multidisciplinary Optimization* 34 (6) (2007) 507–524, ISSN 16151488, 1615147x, doi:10.1007/s00158-007-0105-7.
- [17] G. Pinggen, M. Waidmann, A. Evgrafov, K. Maute, A parametric level-set approach for topology optimization of flow domains, *Structural and Multidisciplinary Optimization* 41 (1) (2010) 117–131, ISSN 16151488, 1615147x, doi:10.1007/s00158-009-0405-1.
- [18] S. Kreissl, G. Pinggen, K. Maute, An explicit level set approach for generalized shape optimization of fluids with the lattice Boltzmann method, *International Journal for Numerical Methods in Fluids* 65 (5) (2011) 496–519, ISSN 10970363, 02712091, doi:10.1002/fld.2193.
- [19] K. Yaji, T. Yamada, M. Yoshino, Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions., *Journal of Computational Physics* 274 (2014) 158.
- [20] M. M. Tekitek, M. Bouzidi, F. Dubois, P. Lallemand, Adjoint lattice Boltzmann equation for parameter identification, *Computers and Fluids* 35 (8-9) (2006) 805–813, ISSN 18790747, 00457930, 10960341, 00426822, doi:10.1016/j.compfluid.2005.07.015.
- [21] M. J. Krause, G. Thaeter, V. Heuveline, Adjoint-based fluid flow control and optimisation with lattice Boltzmann methods, *Computers and Mathematics With Applications* 65 (6) (2013) 945–960, ISSN 18737668, 08981221, 00199958, doi:10.1016/j.camwa.2012.08.007.
- [22] S. Kreissl, G. Pinggen, K. Maute, Topology optimization for unsteady flow, *International Journal for Numerical Methods in Engineering* 87 (13), ISSN 10970207, 00295981, doi:10.1002/nme.3151.
- [23] Y. Deng, Z. Liu, P. Zhang, Y. Liu, Y. Wu, Topology optimization of unsteady incompressible Navier-Stokes flows, *Journal of Computational Physics* 230 (17) (2011) 6688–6708, ISSN 00219991, 10902716, doi:10.1016/j.jcp.2011.05.004.
- [24] K. Yonekura, Y. Kanno, A flow topology optimization method for steady state flow using transient information of flow field solved by lattice Boltzmann method, *Structural and Multidisciplinary Optimization* 51 (1) (2015) 159–172, ISSN 1615147X, 16151488, 1615147x, doi:10.1007/s00158-014-1123-x.
- [25] X. He, L. Luo, A priori derivation of the lattice Boltzmann equation, *Physical Review E* 55 (1997) R6333–R6336, ISSN 10953787, 1063651x, doi:10.1103/PhysRevE.55.R6333.
- [26] X. He, L. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Physical Review E* 56 (6) (1997) 6811–6817, ISSN 10953787, 1063651x.
- [27] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I: Small amplitude processes in charged and neutral one-component systems, *Physical Review* 94 (3), ISSN 0031899x, 15366065.
- [28] X. He, L. Luo, Lattice Boltzmann model for the incompressible Navier-Stokes equation, *Journal of Statistical Physics* 88 (3-4) (1997) 927–944, ISSN 15729613, 00224715.
- [29] S. Chapman, T. Cowling, *The mathematical theory of non-uniform gases*, Cambridge University Press, 1970.
- [30] M. Junk, A. Klar, L. S. Luo, Asymptotic analysis of the lattice Boltzmann equation, *Journal of Computational Physics* 210 (2) (2005) 676–704, ISSN 10902716, 00219991, 10957103, 00219797, doi:10.1016/j.jcp.2005.05.003.
- [31] S. Succi, *The lattice Boltzmann equation for fluid dynamics and beyond*, Oxford University Press, 2001.

- [32] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, *Physics of Fluids* 9 (6) (1997) 1591–1596, ISSN 10897666, 10706631.
- [33] M. Junk, Z. Yang, Outflow boundary conditions for the lattice Boltzmann method, *Progress in Computational Fluid Dynamics* 8 (1-4) (2008) 38–48, ISSN 17415233, 14684349, doi:10.1504/pcfd.2008.018077.
- [34] O. Dardis, J. McCloskey, Lattice Boltzmann scheme with real numbered solid density for the simulation of flow in porous media, *Physical Review E* 54 (4) (1998) 4834–4837, ISSN 10953787, 1063651x.
- [35] O. Dardis, J. McCloskey, Permeability porosity relationships from numerical simulations of fluid flow, *Geophysical Research Letters* 25 (9) (1998) 1471–1474, ISSN 19448007, 00948276.
- [36] G. Liu, M. Geier, Z. Liu, M. Krafczyk, T. Chen, Discrete adjoint sensitivity analysis for fluid flow topology optimization based on the generalized lattice Boltzmann method, *Computers and Mathematics with Applications* 68 (10) (2014) 1374–1392, ISSN 08981221, 18737668.
- [37] A. Griewank, A. Walther, Algorithm 799: Revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation, *ACM Transactions on Mathematical Software* 26 (1) (2000) 19–45, ISSN 15577295, 00983500, doi:10.1145/347837.347846.
- [38] Q. Wang, P. Moin, G. Iaccarino, Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation, *Siam Journal on Scientific Computing*, *Siam J. Sci. Comput* 31 (4) (2009) 2549–2567, ISSN 10957197, 10648275, doi:10.1137/080727890.
- [39] J. Guest, J. Prevost, T. Belytschko, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, *International Journal for Numerical Methods in Engineering* 61 (2) (2004) 238–254, ISSN 10970207, 00295981, doi:10.1002/nme.1064.
- [40] O. Sigmund, Morphology-based black and white filters for topology optimization, *Structural and Multidisciplinary Optimization* 33 (4-5) (2007) 401–424, ISSN 16151488, 1615147x, doi:10.1007/s00158-006-0087-x.
- [41] S. Xu, Y. Cai, G. Cheng, Volume preserving nonlinear density filter based on heaviside functions, *Structural and Multidisciplinary Optimization* 41 (4) (2010) 495–505, ISSN 16151488, 1615147x, doi:10.1007/s00158-009-0452-7.
- [42] F. Wang, B. S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, *Structural and Multidisciplinary Optimization* 43 (6) (2011) 767–784, ISSN 16151488, 1615147x, doi:10.1007/s00158-010-0602-y.
- [43] M. Zhou, B. S. Lazarov, F. Wang, O. Sigmund, Minimum length scale in topology optimization by geometric constraints, *Computer Methods in Applied Mechanics and Engineering* 293 (2015) 266–282, ISSN 18792138, 00457825, doi:10.1016/j.cma.2015.05.003.
- [44] K. Svanberg, The method of moving asymptotes - a new method for structural optimization, *International Journal for Numerical Methods in Engineering* 24 (2) (1987) 359–373, ISSN 10970207, 00295981.